AFRL-IF-RS-TM-2006-4
In-House Technical Memorandum
May 2006

A MATLAB-BASED OZTURK ALGORITHM IMPLEMENTATION

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TM-2006-4 has been reviewed and is approved for publication.

APPROVED: /s/

GERALD C. NETHERCOTT
Chief, Multi-Sensor Exploitation Branch

FOR THE DIRECTOR: /s/

JOSEPH CAMERA
Chief, Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MAY 2006	In-House Final Technical Memorandum May 05- Aug 05

4. TITLE AND SUBTITLE
A MATLAB-BASED OZTURK ALGORITHM IMPLEMENTATION

5. FUNDING NUMBERS
C - N/A
PE - 62702F
PR - 459E
TA - PR
WU - OJ

6. AUTHOR(S)
Steven Salerno

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEC
525 Brooks Road
Rome New York 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEC
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TM-2006-4

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Andrew Noga/IFEC/(315) 330-2270/ Andrew.Noga@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*
A method of determining the underlying statistical distribution from small data sample sets is reviewed. In particular, a Matlab graphical interface is used to allow for testing and comparison to simple histogram techniques. The rank-ordered Ozturk method is seen to perform well for uncorrelated samples

14. SUBJECT TERMS
Ozturk, rank-ordered statistics

15. NUMBER OF PAGES
37

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

1.0 Introduction

This report covers work performed over the summer of 2005 as a summer engineering aide. Work in the applications of ranked-ordered statistics as a method of signal data analysis was conducted. More specifically work was conducted on the Ozturk algorithm, which is considered to have demonstrated superior performance over histogram and simple parameter-based techniques [1-4]. The work includes the construction of a graphical user interface that utilizes the algorithm's power to both identify and analyze probability density functions.

1.1 Overview of Ozturk Algorithm

The Ozturk algorithm is an identification algorithm that allows for a variety of tests that not only identify a PDF, but also analyzes the sample data. The core algorithm was developed by Aydin Ozturk [4], and further worked upon in concurrence with E.J. Dudewicz [2]. Prior work was carried out with the help of Syracuse University and the only documented implementation of the algorithm for real world use was associated with research conducted at the AFRL Sensors Directorate at the Rome Research Site.

The basis of the algorithm deals with constructing a null-linked graph of vectors from the ordered statistics of the sample data. The algorithm is scale and location invariant, which makes it particularly attractive. From this graph one can easily identify a distribution based on its endpoint, Q_n, and surrounding null points. From the ordered statistics other tests can be carried out, including but not limiting to, testing for Normality, and parameter estimation. The power of the test comes from the ability to use small sample sizes while still being able to correctly identify the best fit distribution. The entirety of the algorithm will be discussed in Section 2.

1.2 Summer Overview

During the 2005 summer a Matlab program was developed to be able to evaluate and demonstrate the Ozturk algorithm. Using the computing technology we have now we were able to recreate the two null data charts using Monte Carlo methods that were originally calculated using numerical integration. With these charts and the equations derived for the algorithm, the program could match all the results that were provided by samples in [2], and [3]. Along with the actual implementation of the algorithm and associated GUI, one further extension of the algorithm was created in order to extend the features of the code. The GUI prototype will be further discussed in Section 3.

2.0 PDF Identification

Currently there exists an abundance of tests in order to determine/analyze a PDF. Histogram techniques depend on the number of bins, which is specified by the user, and the number of samples, given by user. It can be shown however that these techniques can cause great amounts of error on both sides of the extrema. A low number of samples can and usually will result in a histogram with no known shape as in Figure 2-1.

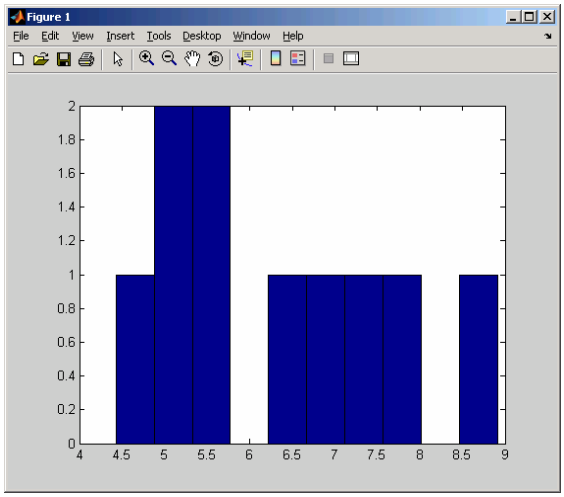


Fig 2-1 Histogram with low sample count

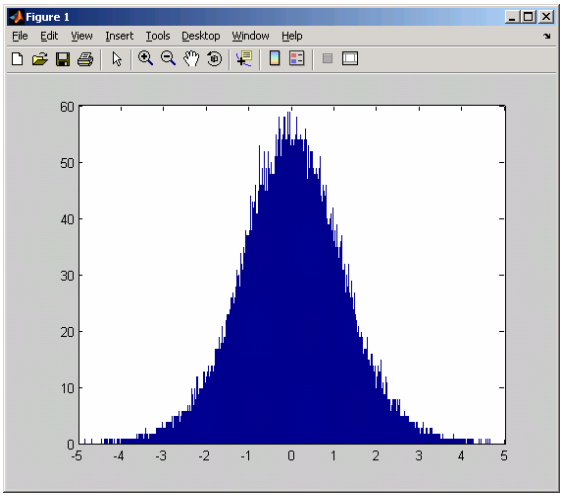


Fig 2-2 Histogram with high bin count

Changing the bin count has does not correct this problem because from this figure we cannot easily define a known PDF from the histogram. On the other hand if the sample size and bin count are high, then the histogram becomes jagged in shape, as in Figure 2-2, instead of being smooth. Even though the shape isn't perfect, the distribution can still be determined with a high confidence. Note that the histogram does not use a rank ordering of the data samples.

Other well known techniques to identify PDFs include the Chi-Square goodness-of-fit test, Kolmogorov-Smirnov goodness-of-fit test, Anderson-Darling test, Shapiro-Wilk test, and the quantile-quantile plot, [7]. The Chi-Square goodness-of-fit test is applied to data that has been binned, such as a histogram. From this the test can decide from what specific univariate distribution the sample came from. However the test is sensitive to the number of bins, which are specified by the user. Also it is dependent on how the data has been put into its bins, and requires a sufficient sample size to provide for a satisfactory solution.

The K-S test (Kolmogorov-Smirnov test) also decides whether a sample comes from a specific distribution. Using rank-ordered data and the empirical distribution function a graph can be constructed as shown in Figure 2-3, from which we can estimate the test statistic. The maximum distance between the two curves is calculated and the

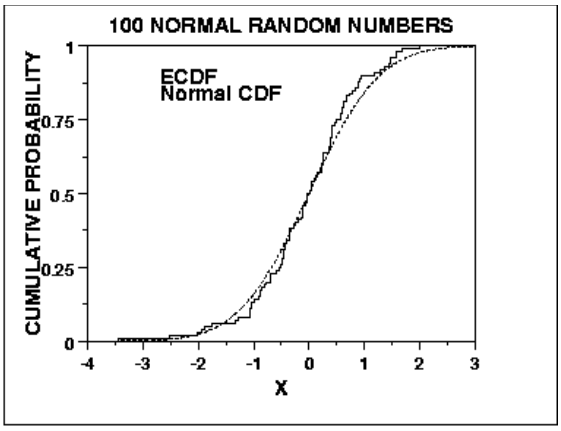


Fig 2-3 Normal CDF for 100 random samples[7]

solution is found as the closest fit to a set of known values. Advantages to the K-S test include that it can be done with any sample size, and it does not rely on the underlying CDF being tested. However the K-S test cannot be applied to discontinuous functions, it is more sensitive towards the middle of the distribution, and the parameters of the sample must be known, or the test is no longer valid. A solution to the last disadvantage comes from Lilliefors test for Normality, an extension of the K-S test. In this test the sample mean and standard deviation are calculated and the Z-score is found. With the Z-score the empirical CDF is calculated, plotted and the maximum distance is found as the test statistic.

The Anderson-Darling test is a modified version of the K-S test. It improves upon the sensitivity of the center of the K-S test by weighting the tails. Also the Anderson-Darling test makes use of the specific distribution being tested to gain the critical values, which in turn makes a more sensitive test overall. However calculating the critical values is a disadvantage, and for some distributions can be undefined.

Unlike the other tests described thus far, the Shapiro-Wilk test only tests a sample for Normality. However the power and sensitivity is much higher because of this. The test calculates a test statistic from which the value can identify whether the sample is Normal or not.

The Q-Q plot (quantile-quantile plot) is a graphical means of interpreting whether two distributions come from the same family. This can both identify a specific distribution if one is known and the other is unknown, or can determine if both unknown/known samples come from the same distribution. The Q-Q plot takes the values of the quantiles of each distribution and plots them against each other with a 45-degree reference line as shown in Figure 2-4. The sample sizes do not have to be large,

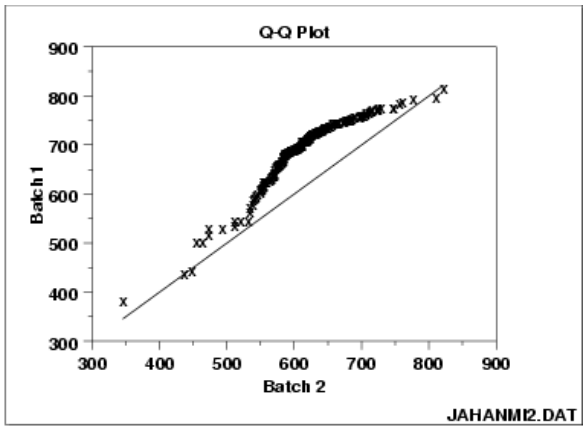


Fig 2-4 Sample Q-Q plot[7]

but the bigger the sample size the more accurately the correlation can be portrayed. The two distributions do not have to be equal in size. Also many properties of a distribution can be tested all at once, including shifts in location and scale, changes in symmetry, the presence of outliers, etc.

2.1 Ozturk Algorithm

All of the aforementioned tests are reliable in the sense that you are testing for a specific distribution. Often the test is for normality (Gaussian PDF is assumed). However when the solution does not match the hypothesis no alternative possibility is given. Also a majority of the tests start to lose their power when the sample size decreases. Both these issues are addressed through the Ozturk algorithm [1-4]. The algorithm uses the sample statistics to create "linked vectors' to create a goodness-of-fit test for univariate and multivariate distributions. The only foreseen disadvantage the algorithm has is that for larger sample sizes the computational time is slowed dramatically. In this case, the other tests previously mentioned can identify and analyze distributions with a high confidence level at high sample sizes.

2.2 Algorithm Procedure

General reference is made to [2]. Let $X_{1:n} \le X_{2:n} \le \ldots \le X_{n:n}$ be the sample ordered observations from a distribution with the CDF $F(x;\mu, \sigma) = F\{(x- \mu)/\sigma\}$ where μ and σ are the location and scale parameters, respectively. For our case we use the Normal distribution as the null hypothesis; however any distribution could be used as the null.

For illustration we will assume that $X_{1:n} \le X_{2:n} \le \ldots \le X_{n:n}$ is from a Normal distribution with an unknown mean μ and unknown variance σ^2. To make the sample linked vectors we need two main components, the length of the ι^{th} vector, and the angle to the horizontal axis θ. Let

$$Y_i = (X_i - \overline{X})/S \tag{2-1}$$

where $\overline{X} = \sum X_i / n$ and $S = \{\sum (X_i - \overline{X})^2/(n-1)\}^{1/2}$. $Y_{i:n}$ is now considered the length of the vectors. Now let $m_{i:n}$, $m_{2:n}$, …, $m_{n:n}$ be equivalent to the expected order statistics of the standard Normal distribution, $m_{i:n} = E((X_{i:n}- \mu)/\sigma)$. The angle horizontal to the x-axis is $\theta_\iota = \pi\Phi(m_{i:n})$ where $\pi = 3.14159\ldots$ and

$$\Phi(m_{i:n}) = \int_{-\infty}^{m_{i:n}} (2\pi)^{-1/2} e^{(-t^2/2)} dt\,. \tag{2-2}$$

Since $m_{1:n} < m_{2:n} < \ldots < m_{n:n}$ and $0 < \Phi(m_{i:n}) < 1$ then $0 < \theta_1 < \theta_2 < \ldots < \theta_n < \pi$. $\theta_{i:n}$ is now considered the angle between the x-axis and the ι^{th} vector.

To obtain the sample linked vectors we use the points $Q_i = (U_i,V_i)$, where $U_i = (\sum_{j=1}^{i} |Y_{j:n}| \mathrm{Cos}\theta_j)/n$ and $V_i = (\sum_{j=1}^{i} |Y_{j:n}| \mathrm{Sin}\theta_j)/n$ are the coordinates of Q_ι (ι = 1,…,n) and $Q_0 = (0,0)$. U_i and V_i can then be defined as the test statistics, as following

$$U_n = 1/n \sum_{i=1}^{n} |Y_{i:n}| \mathrm{Cos}\theta_i\,, \tag{2-3}$$

$$V_n = 1/n \sum_{i=1}^{n} |Y_{i:n}| \mathrm{Sin}\theta_i\,. \tag{2-4}$$

To obtain the null linked vectors to construct the "known" of the null hypothesis $m'_{i:n} = E(|Y_{i:n}|)$, and $E(m_{i:n})$ are used. To obtain the expected values numerical integration must be used.

$$E(x_{k|n}) = \frac{n!}{(n-k)!(k-1)!}\int_{-\infty}^{\infty} x[\frac{1}{2} - \Phi(x)]^{k-1}[\frac{1}{2} + \Phi(x)]^{n-k}\,\phi(x)dx, \qquad (2\text{-}5)$$

$$\text{where } \phi(x) = (2\pi)^{-1/2} e^{-x^2/2} \text{ and } \Phi(x) = \int_0^x \phi(x)dx.$$

Using the expected values of $m_{i:n}$ and $Y_{i:n}$ we obtain the point $Q_n = (U_n, V_n)$ which becomes the null-linked vectors endpoint that is the "known", other extensions use this information for analysis.

In a sample that we assume to be Normal, we would expect that the sample linked vectors would closely follow the trajectory of the null-linked vectors, as in Figure 2-5.

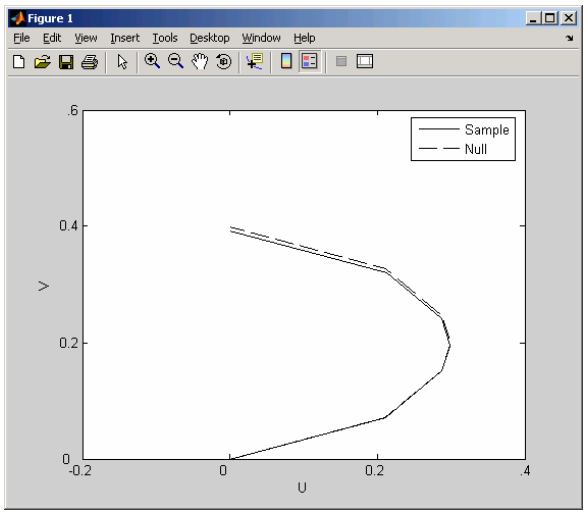


Fig 2-5 Vector Chart for Normal (n=6)

However if the sample is not Normal the two paths should differ; how much they differ by depends on the sample distribution. This use of the algorithm allows for an *ad hoc* version (visual inspection) to be used and allow for distribution identification. Although this can be done, it is rarely used because we have the test statistics that are provided by the algorithm to more powerfully identify what a sample distribution's true distribution is.

A group of extensions have been developed for the algorithm to gather more information about both the distribution and the test statistics. Using $m'_{i:n} = E(|Y_{i:n}|)$, we define the expected value for Q_n for the Normal distribution to be $E(Q_n)$ =

$(0,(1/n)\sum m'_{i:n}\,\mathrm{Sin}\theta_i)$. Symmetric identities for the Normal distribution can be employed to see that $E(U) = 0$, and a model can be fitted to V to obtain

$$E(V_n) = 0.326601 + 0.412921/n\,. \tag{2-6}$$

Also using more properties of the test statistics and the linked vectors we can obtain models for the variances of U and V

$$\mathrm{Var}(U_n) = \sigma_u^2 = 0.02123/n + 0.01765/n^2, \tag{2-7}$$

$$\mathrm{Var}(V_n) = \sigma_v^2 = 0.04427/n - 0.0951/n^2, \tag{2-8}$$

which will help in further analysis tests. Both U and V are distributions, more specifically Normal for large values of n. Since Qn is the joint distribution of U and V, a test statistic for a bivariate Normal distribution can be found and simplified as

$$\frac{u^2}{\sigma_u^2} + \frac{(v-u_v)^2}{\sigma_v^2} = -2\ln\alpha\,, \tag{2-9}$$

where α is the confidence level. Using α we can get the 100(1- α)% confidence ellipses, which can be used to help test for Normality, as shown in Figure 2-6. We can then use

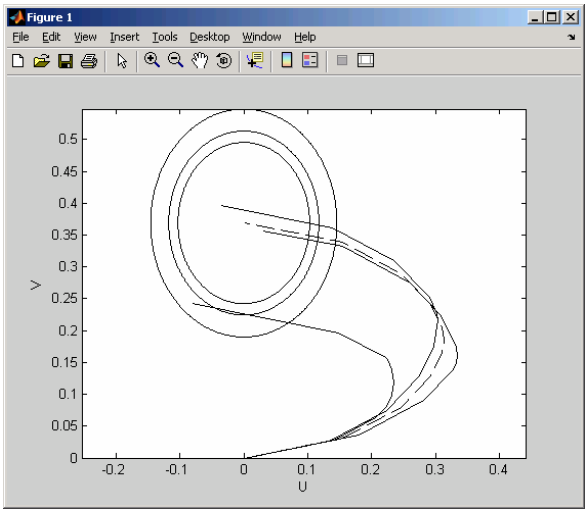


Fig 2-6 Sample Data with Confidence Ellipses

α as a measure of whether or not to accept or reject the answer given by the test statistic.

One of the most powerful aspects of the test is that it provides within itself the ability to do parameter estimation based on the test statistics. Using the statistics $T_1 = \sum X_{i:n}\mathrm{Cos}\{\pi\, \mathrm{F}_0(m_{i:n})\}$ and $T_2 = \sum X_{i:n}\mathrm{Sin}\{\pi\, \mathrm{F}_0(m_{i:n})\}$ we can estimate the location and scale. For better estimates we can use

$$\hat{\mu} = \sum_{i=1}^{n} (\mathrm{Sin}\,\theta_i) X_{i:n} / c\,, \tag{2-10}$$

$$\hat{\sigma} = \sum_{i=1}^{n} (\mathrm{Cos}\,\theta_i) X_{i:n} / b\,, \tag{2-11}$$

where $c = \sum \mathrm{Sin}\,\theta_i$ and $b = \sum \mu_{i:n}\mathrm{Cos}\,\theta_i$.

The shape parameter of a distribution can also be calculated but was not done here because no distributions with shape parameters were analyzed in this time frame of this research.

2.3 Problems Encountered

While working with the algorithm there were a few situations in which interpretations of wording came up and had to be addressed. Also some information was obscure and had to be found by other means. The first major problem uncovered was that the only identification chart that could be found was in [1]. However this chart, using their own values calculated by their models, shows this chart to be incorrect value wise. Although the values for each of the points, and lines may be wrong, the locations seemed to be preserved. In this case we were able to use their chart as a known to test against ours, by comparing only relative locations and not values.

Next we were given the $E(|\mathrm{Y}_{i:n}|)$ in [2], but were not given any indication of what the $\mathrm{m}_{i:n}$ values were, or the means to regenerate either set. The first option was to find the equation (2-5) for the numerical integration and evaluate it to obtain the correct values. However due to the complexity that numerical integration sometimes presents we chose a Monte Carlo method. 1,000,000 runs of sample sizes n = 6, 10, 20, 50, and 100 were used to generate the values that the numerical integration would. As a result the error in our answers compared to the chart answers in [5] was less then 0.0001. This

error is acceptable because through an experiment the values of U and V do not vary greatly unless the error is above 0.01.

A problem also occurred with the wording regarding α and the "confidence ellipses." It is understood that the 100(1- α)% level is the level at which the user is that confident. For example, in Figure 2-6, the middle ellipse is the "95% confidence ellipse." However it is actually saying there is a 95% probability of it not being Normal instead of being 95% confident it is Normal. As you get closer to the null endpoint the confidence in the answer should continue to go up, until the null endpoint is hit and the confidence should be 100%. If we used this interpretation, it says that at the null endpoint the confidence ellipse is 0%. To resolve this wording issue, confidence ellipse could either be considered as more of an error ellipse, or the equation could be changed to 100(α)%.

Another problem with maximizing performance deals with the fact that the algorithm actually has 9 different ways to standardize the sample order statistics. In this paper and the experiments presented we use what is called the *Type 1* version or $Q_n^{(1)}$ [3]. This version is used because it presents the best power to identify and analyze over the most distributions. However if we were to use each of the versions to identify and analyze we would be able to increase the accuracy of the algorithm even further. This is because each of the versions is sensitive to a certain distribution, and thus more powerful for those distributions.

3.0 GUI Implementation

The entire GUI and code was built from the ground up for the sole purpose of PDF identification without specific application in mind. All equations for the algorithm, PDF information, and other likely equations were gathered and stored in multiple functions. Matlab was to be the primary focus, while using Octave, a Matlab look alike, to provide sample data. However through the Mathworks community file sharing a Matlab function was found to generate data for all needed distributions. The GUI is constructed of the main window where all functions takes place, along with 4 smaller windows that only display data.

3.1 Main Window – Distribution

The main window is featured in Figure 3-1. The main window is divided into 3

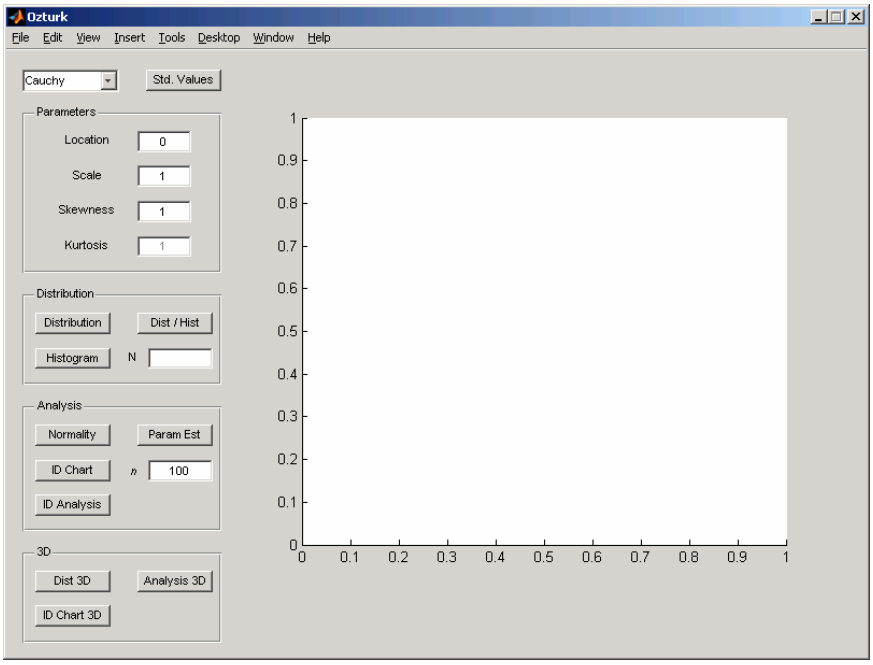


Fig 3-1 GUI Main Window

subsections, Distribution, Analysis, and 3D. In the top left corner is a pull down menu that lets the user select which distribution they would like to work on, as seen in Figure 3.-2.

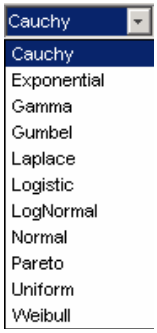


Fig 3-2 Distribution List

There are currently 11 different distributions that can be chosen for the first part of the tool. The user simple chooses which of the distributions they would like to begin with.

After the selection of the distribution by the user, the Std. Values pushbutton needs to be pressed. This simply puts the values of the location, scale, skewness and kurtosis parameters to 0, 1, 1, and 1, respectively. These values are the standard values for each of the distributions except for the Pareto distribution which needs to have a location greater than one to be defined, so its standard location value is 1. This is done to reset the data from any previous run of the tool.

Next is the Parameters panel, as in Figure 3-3, which displays to the user the current values of the parameters of the distribution they have chosen. For this scenario we will be using the Normal distribution throughout the discussion of the GUI. If the distribution is not dependent upon a parameter it becomes grayed out and unable to be change. Also if a user decides that they would like to see the distribution with parameters other than the standard values, they can simply change them.

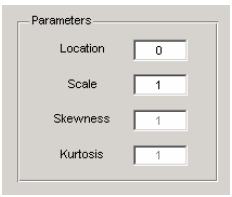


Fig 3-3 GUI Parameters Panel

The first set of tools that can be used are contained in the Distribution panel, as in Figure 3-4. Within the panel are 3 pushbuttons and an edit text area. The Distribution pushbutton takes the type of distribution selected by the user, and the parameters selected

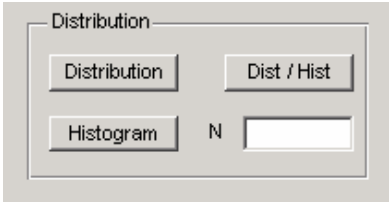


Fig 3-4 GUI Distribution Panel

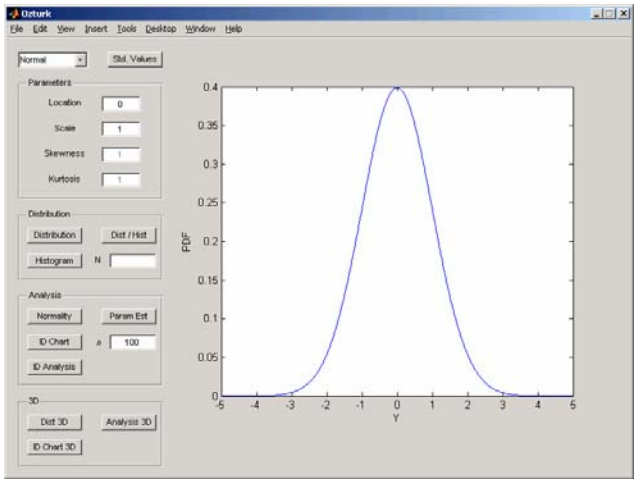


Fig 3-5 GUI Distribution Plot

by the user, and plots the PDF on the graph on the right hand side of the GUI, as seen in Figure 3-5.

The Histogram pushbutton is similar in that it takes the selected distribution and the set parameters. It plots a histogram of 1,000,000 samples, as shown in Figure 3-6.

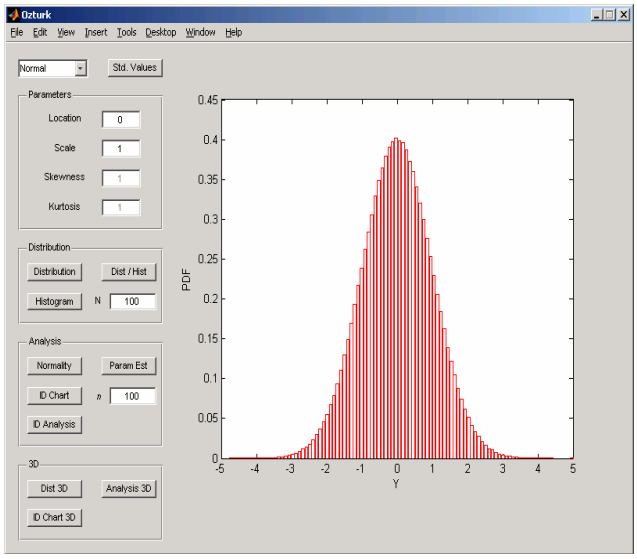


Fig 3-6 GUI Histogram Plot

The Histogram is also ruled by another parameter which is the number of bins to sort the data into. This parameter can be defined by the user as was the parameters of the distribution. The edit text box labeled N is where the user is allowed to make changes to the number of bins. For Figure 3-6 the current N is equal to 100, however we can change

that to any number we choose, as long as it is a positive integer. Changing the number of bins allows the user to maximize the power of the histogram by choosing the optimal number of bins.

The last tool in the group is the Dist/Hist pushbutton. This first plots the histogram of the distribution with the selected parameters and bins, and then plots the distribution over it, as in Figure 3-7. However this tool can only be used after the

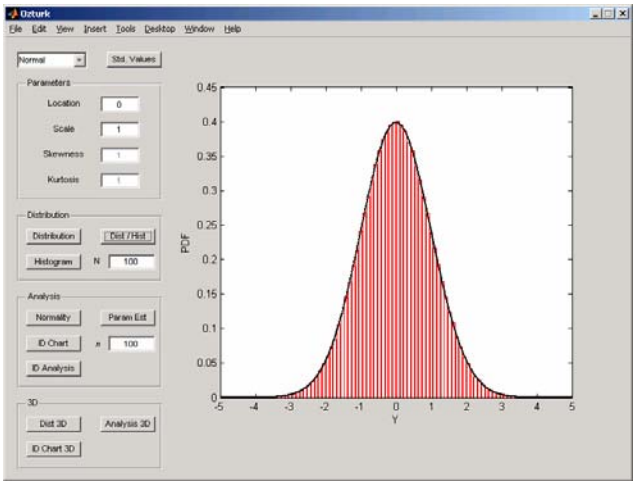


Fig 3-7 GUI Dist/Hist Plot

Histogram pushbutton has been used because it reuses the data from the Histogram tool instead of creating another different set. For larger bin sizes there would be no large difference if two data sets were constructed, though as bin sizes decrease the appearance of the Histogram begins to vary more.

This tool is not a necessary part of the GUI since its function has nothing to do with the Ozturk algorithm. However this can be used more so as a learning feature to be able to identify the different distributions. Also if a user has a known shape and known parameters but does not know the distribution it is, this could become a potential identification tool.

3.2.0 Main Window – Analysis

The Analysis panel contains the main tools that have been defined in the Ozturk algorithm, as shown in Figure 3-8. It contains four different pushbuttons that are all functions of the algorithm, and an edit text box for the sample size of the data. There are

a few restrictions in place as to how to operate within this panel. First when testing a sample, the data must be given in a .mat file and the variable of the data must be call x_userdata. Also within the data, if there is more than one data set that is given in the variable they must be arranged by having the first data set in the first column, the second set in the second column, etc. With regard to the direction of the pushbuttons, the Normality pushbutton must be first because it loads the sample data from the source then saves it to another file from which the other pushbuttons can use it from.

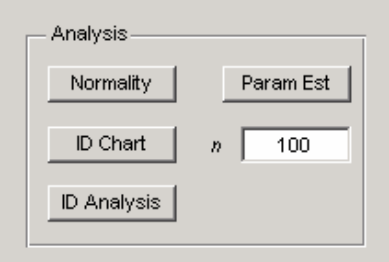


Fig 3-8 GUI Analysis Panel

For our purposes we will use a data set that has been given in [2], which contains 3 data sets, two generated by a Normal distribution and the third by an Exponential.

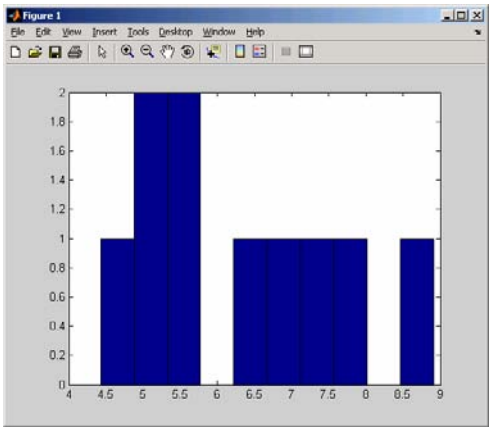
Sample 1: 5.62 6.76 7.90 5.77 7.51 4.92 4.43 6.51 5.30 8.91

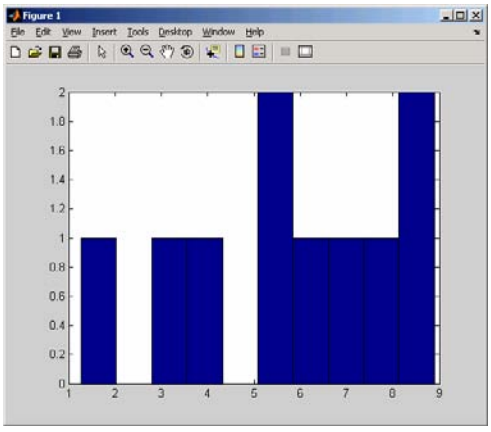
Sample 2: 7.43 5.55 2.91 4.17 1.27 5.44 8.89 8.75 6.20 7.14

Sample 3: 8.51 7.12 21.49 7.15 6.94 12.97 7.30 0.19 7.20 4.18

Preliminary tests were run on the data sets to see if their true distributions could be found. All tests proved inconsistent and could not tell us the true distribution, Figure 3-9 – 3-11.

Fig 3-9 Sample 1 Histogram





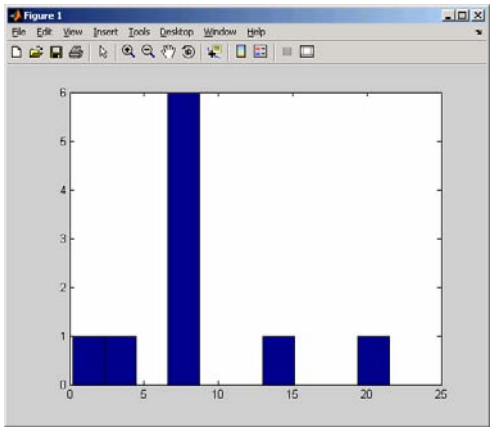


Fig 3-10 Sample 2 Histogram

Fig 3-11 Sample 3 Histogram

The Normality pushbutton runs the Ozturk algorithm. After pressing the pushbutton the user is prompted to pick which file they would like to load into the GUI for testing, as in Figure 3-12. After the selection has been chosen the data is put through

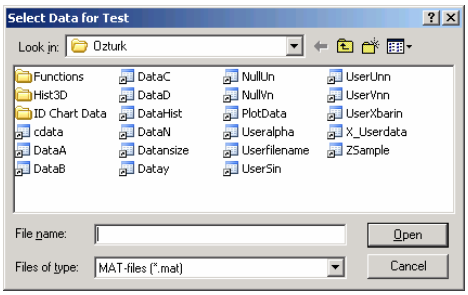


Fig 3-12 GUI Data Selection Window

the algorithm one data set at a time. The null-linked vector is plotted, along with the sample vectors for each of the sample data sets, and the confidence ellipses are drawn, as shown in Figure 3-13.

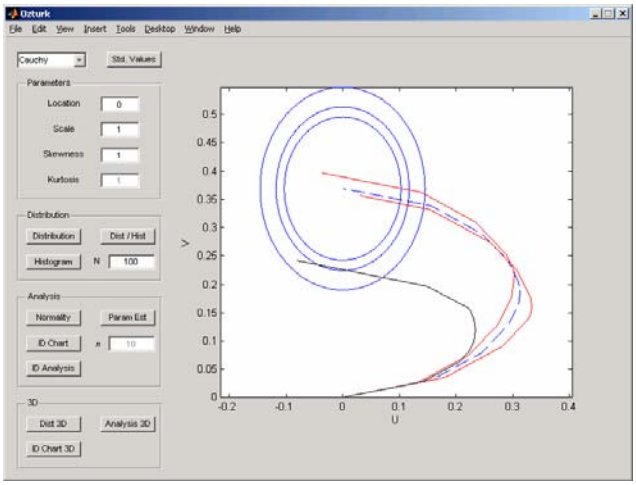


Fig 3-13 GUI Normality Test Plot

Within this test we can gain valuable information about the sample data with what little information we have about the sample data already. First the null-linked vector is plotted as a blue dotted line. Next any line that falls inside of the 95% confidence ellipse is plotted in red, and any line that falls outside of that ellipse is plotted in black. Within the edit text box we can now see the sample size of the data; in this case the data vectors are all 10 numbers long. From this graph we are able to identify which two samples are from Normal distributions and the one from an Exponential distribution. More generally if a normality test fails or is in question we would still like to automatically determine the underlying distribution.

3.2.1 User Data Window

One of the functions within the Normality test is the ability to display information about each of the data samples. To do this a user has to click on the sample linked vectors that they wish to view the data on. This click will open another smaller GUI that displays the mean, standard deviation, α value, U_n, V_n and filename from which the data

sample was taken from, as in Figure 3-14. Then sample mean and standard deviation are found using $\overline{X} = \sum X_i / n$ and $S = \{\sum (X_i - \overline{X})^2 / (n-1)\}^{1/2}$ instead of the parameter estimation techniques, and the α value is calculated from 2-14.

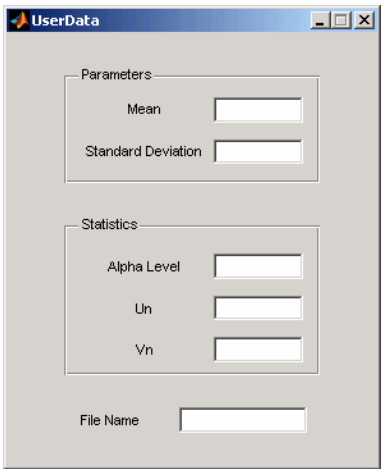


Fig 3-14 User Data Window

3.2.2 Identification Chart

The basic idea behind the ID chart is that if the data isn't Normal then the endpoint is going to fall somewhere away from the Normal endpoint. Using data from other generators to simulate other distributions a library of points of these distributions, based on the reference distribution (the Normal) can be formed on a plot. This set of predefined points for a given sample size is known as the ID chart. For each size of n the ID chart must be redefined because each n has a specific endpoint that is calculated. This is the heart of the Ozturk algorithm in that it can identify the distribution with the sample data set it has in just one pass. When the sample vector is plotted, the "closest" point to the endpoint is considered to be the true distribution for that data. The more distributions that are plotted onto the ID chart the higher the power of the chart in identifying the different distributions although more overlap may occur.

Using the chart from [1] as an example we notice that there are both points and lines on the ID chart. This is associated with the properties of the algorithm. Since the algorithm is scale and location invariant, there are no variables for certain distributions so

these plots of the location-scale families become points on the chart. However there are those families that have a shape parameter to them such as skewness or kurtosis. When the distribution has one shape factor it then has one variable and a line is plotted onto the chart instead of a point. If a distribution has two shape factors it plots a series of lines, instead of one.

Again using 1,000,000 runs in a MC simulation we were able to obtain the $E(|Y_{i:n}|)$ for six distributions; Cauchy, Exponential, Laplace, Logistic, Normal, and Uniform. The other three location-scale distributions, Pareto, Gumbel, and LogNormal were not plotted because of difficulties regarding position and shape on the ID chart. The last two distributions, Weibull, and Gamma were not plotted because they have one shape parameter and time could not be allocated to finish the amount of simulations needed to create the line with an acceptable accuracy. Also MC simulations of n = 6, 10, 20, 50, and 100 were used to limit time and resources needed create any more sample sizes.

As a result we were able to correctly place the first six distributions on the chart, and create a known ID chart for the five sample sizes. Within the GUI the user can then select which of the five charts they would like to look at from the ID Chart pushbutton, as shown in Figure 3-15.

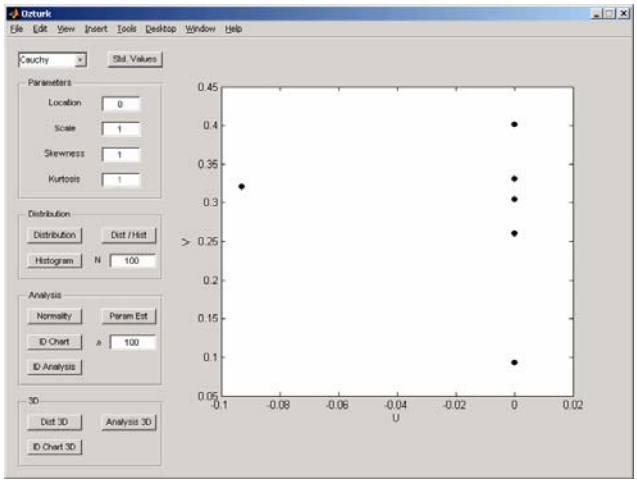


Fig 3-15 GUI ID Chart Plot for n = 100

3.2.3 Distribution Type Window

In an effort to give more information to the user another window was created to show the user the distribution type and endpoint value for the point that they click on,

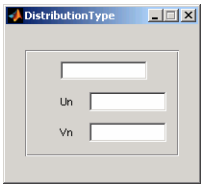


Fig 3-16 Distribution Type Window

shown in Figure 3-16. This allows the user to differentiate between the points on the ID chart, and also know what the endpoint values are for that specified n value.

3.2.4 ID Analysis

The analysis tool, whose function it is to return which distribution fits the sample data best, can be accomplished in two ways. This section will explain the way that could be used that follows the algorithm. The other will be discussed in the section concerning the 3D ID analysis. The user window that is created within this pushbutton is also described later in this section.

After the ID charts have been created and saved the analysis of the sample data can occur. First the sample data is plotted onto the ID chart of the corresponding sample n size, as seen in Figure 3-17. Also in the edit text box the size of the sample data is

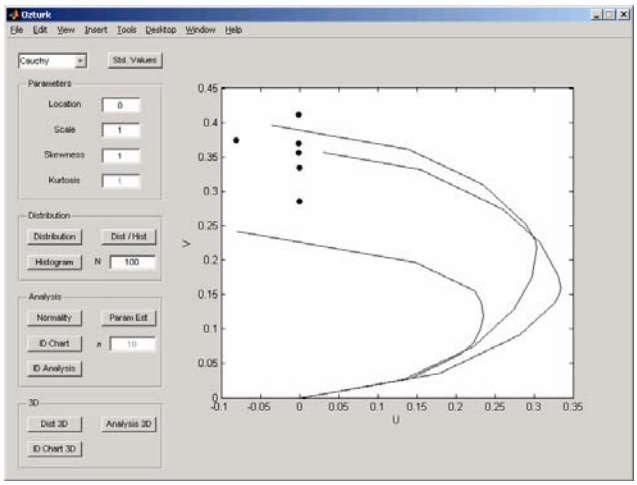


Fig 3-17 GUI ID Analysis Test Plot

placed on the figure. This allows for the user to visually determine which of the distributions best fits the sample data. This method can be very inaccurate when the data set comes close to multiple ID points.

The method that the algorithm provides for is using the α values to determine which distribution best fits the data (this method was not used in this code because the new extension that was conceived provided its own method of identification). Using equation (2-9) we can calculate the α values for the Normal distribution. In order to find the α values for all of the distributions we would have to generalize the equation. Once generalized we would be able to obtain α values for all the distributions, and with the α values sort them so the highest value would be the true distribution for that sample data. This method seems feasible; in [3] there is a plot that has confidence ellipses for the Exponential distribution.

3.2.5 Parameter Estimation

Another extension that the algorithm lends itself to is parameter estimation. Because the algorithm can use its own statistics, methods such as moments do not have to be calculated for each of the distributions. Using the test statistics T_1 and T_2 the parameters of the sample data for any distribution can be found. This is the technique that is used for parameter estimation in the code. When the Param Est button is pressed the sample vectors are plotted onto the ID chart of the corresponding size, as shown in Figure 3-18. If the button is pressed after the ID Analysis button then there will be no

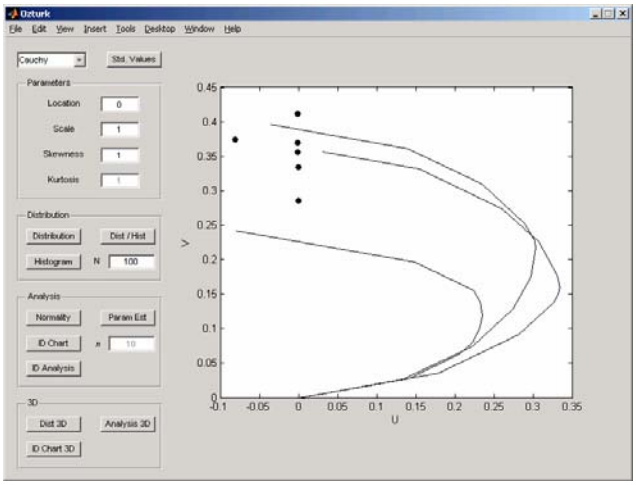


Fig 3-18 GUI Parameter Estimation Plot

change in appearance of the GUI. However now when a user clicks a the sample vector a window comes up with the values of its mean, standard deviation, and shape if it has one, as shown in Figure 3-19. These values may vary from those that are displayed in the

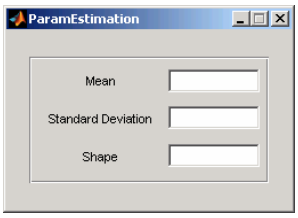


Fig 3-19 Parameter Estimation Window

User Data window. This is because the User Data window is taking the sample mean and sample standard deviation. The sample mean and sample standard deviation will not

exactly match the true mean and true standard deviation, but should be somewhere near it in value.

3.3.0 Main Window – 3D

The last set of tools that can be used is included in the 3D panel of the main GUI, as in Figure 3-20. This extension of the algorithm was created this summer in order to increase performance and enhance the plots associated with the algorithm. In order to even begin the process a new Matlab function had to be created to construct a 3D (technically 2D) histogram. This is accomplished by selecting a grid and creating a sliding window that goes over each grid one at a time. It then gets all the values that fit within that grid, and the number of values then becomes the height at that point on the grid. The function was generalized for other applications.

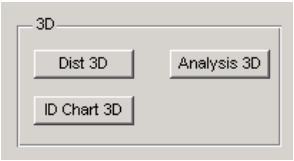


Fig 3-20 GUI 3D Panel

3.3.1 Distribution 3D

With the new histogram function we are now able to take the endpoint data and create a histogram around its null endpoint. The function to generate the $Y_{i:n}$ values was run 50,000 times while storing all the values. Each of these sets was put into another function to get the final endpoint of that specific run and each of the endpoints were saved. Then the endpoints were put into the new histogram function to obtain a graph that resembles a mountain, as in Figure 3-21. Each distribution has a different height and shape, which allows for some difference in appearance. The plot represents the distribution of the endpoints around the null. Also if we were to look straight down the Z-axis we should be able to somewhat see the confidence ellipses.

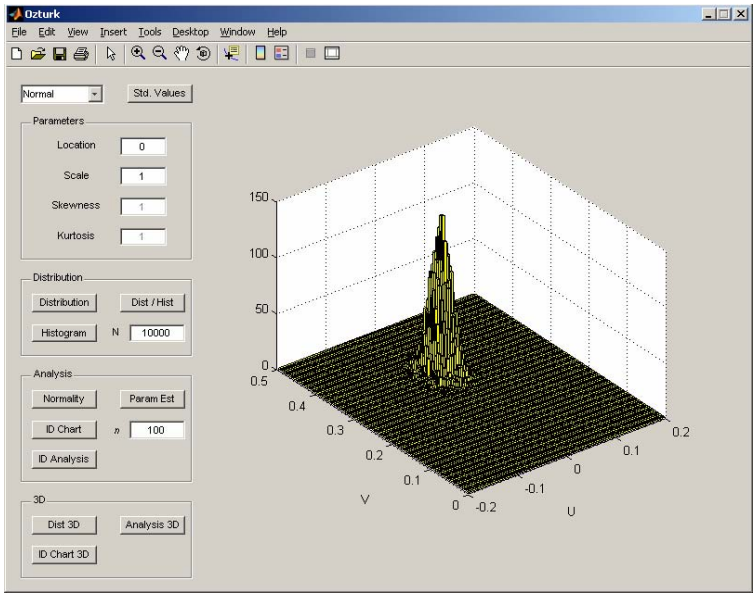


Fig 3-21 GUI Dist 3D Normal n = 100 Plot

3.3.2 Identification Chart 3D

As with the ID chart we can make a new 3D ID chart of these distributions because they are all based on their endpoints. After generating all the distribution charts for a given n we can plot them all onto one chart because they all share the same grid, as in Figure 3-22. Each distribution is now color coded in order to easily differentiate

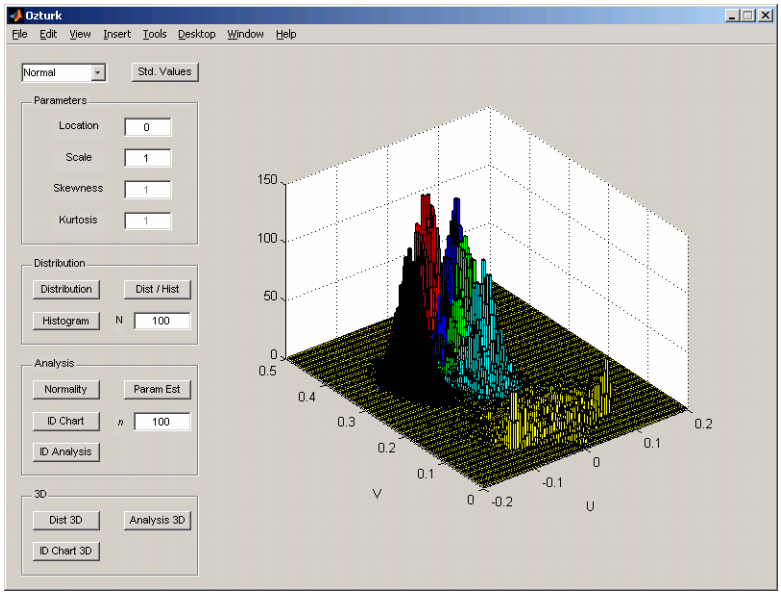


Fig 3-22 GUI ID Dist 3D n = 100 Plot

between two distributions. Normal is blue, Uniform is red, Exponential is black, Laplace is cyan, Logistic is green and Cauchy is yellow. We can now use these ID charts as we did with the other ones, by plotting sample data onto the charts and determining the best distribution for the data.

3.3.3 ID Analysis 3D

Using the same method above as for the 2D analysis we can plot our sample data onto the ID chart of the corresponding size, as seen in 3-23. The data is plotted as white "flag poles" that extend above the current ID charts. Their position is determined the same way that the distributions are plotted, except their height is multiplied by a factor related to the sample size to make sure they are higher than the distributions.

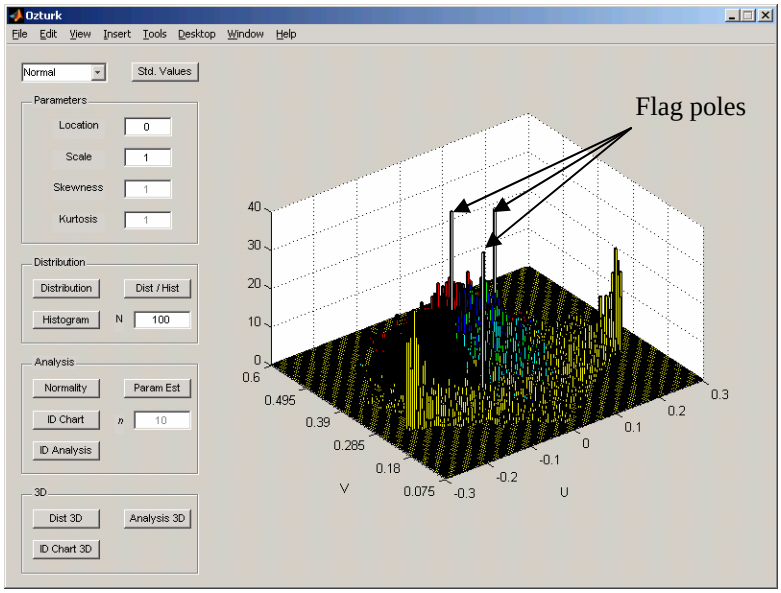


Fig 3-23 GUI Analysis 3D Plot

Using these locations we can now provide our own analysis of the data to determine the true distribution. This is the analytic method that we use to identify the distributions through the ID Analysis pushbutton. First, the locations for each of the data

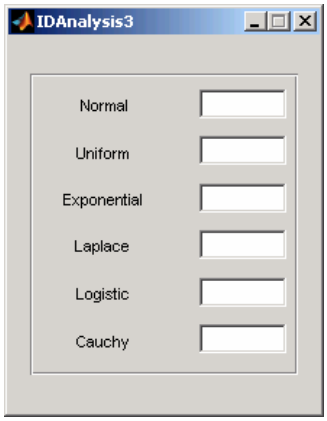


Fig 3-24 ID Analysis Window

samples is found and recorded. We then go back through for each sample and record the heights of each distribution at that location. These values are put into the ID Analysis window of Figure 3-24. Based on the values for each distribution you can tell which should be the true distribution, which would be the one associated with the highest value. This method is less complex because we take the endpoints which are already given and find their location on the grid.

4.0 Conclusions

Through the use of the Ozturk algorithm we can clearly see its power to identify and analyze distributions with small data sets. However one worry that has surfaced is the possibility that the algorithm hasn't been used, or has been and not published, in the past ten years. This leads to concerns about the algorithm, but thus far no major problems have been incurred with it.

4.1 Applications of the Ozturk Algorithm

The primary purpose of the Ozturk Algorithm seems to be that of addressing the need to analyze and/or identify statistical information. Previous work with the algorithm has been done with radar and IR imaging at the RRS Sensors Directorate [6]. Possible applications include signal processing, video and images (especially IR), chemical detection, Steganalysis, etc.

4.2 Constraints of the Ozturk Algorithm

As stated previously the algorithm is not usually run for high sample sizes because of the processing time that is needed for the calculations. One way to reduce this time is to have the expected values that are needed such as for the $Y_{i:n}$ and $m_{i:n}$ values. However this is not necessary because other tests appear to be just as powerful for identifying and analyzing at high sample sizes.

The new extension that has been created also has some time constraints on it. Monte Carlo runs had to be cut down because of the time needed to reproduce values for the $Y_{i:n}$ values. Each run of the simulation had to be saved which was the reason for the time and memory issues. Then each of those runs must be put into the actual algorithm which takes about half the above time, but is still much longer than running just one averaged sample or a small data set. As a result all of the data must be saved to some point on the computer and the amount of memory that is used drastically slows it down. So both the time and amount of data that is saved are major constraints to how accurate this new extension can be. Currently it seems feasible for our uses, and there are possibilities to reduce this problem in further work. This may simply be a code optimization issue.

4.3 Problems & Bugs

Within the GUI there have been some problems that have either not been able to be fixed or the cause is currently not known for the problem and would take some analysis to troubleshoot.

Currently there is a problem with the Cauchy and Pareto histograms. Both of the PDF equations work and the PDF can be viewed. However both the distributions do not have a histogram to be viewed for some unknown reason. The cause of this has been attributed to the data generators that were used.

Another issue is that the histogram function in Matlab takes the sum of all the data, so when there are a million samples the highest points are close to 50,000 times higher in height. However these values are too high and to fix this problem a weighted factor had to be multiplied by all the histogram heights to get the right value. However once the data has been changed it cannot be reproduced as a histogram, and must be plotted as a bar graph. So when the histogram has a lower number of bins, usually less then 100 for most distributions, there appear to be spaces between the bars. Histograms are consecutive bins; there are no spaces between the bins. Although there are spaces between the bins, they do not seem to affect the shape very much, and for larger bin number these spaces can no longer be seen.

Some other minor flaws in the graphing of the distributions and the histograms are the window size. At first when the scale is increasing the distribution quickly changes. However as the values get much higher the shape begins to change very slowly. So the window size deals mainly with the quick changes. Although the changes are always in the window then as the scale increases the window size does also. This is what is wanted to a point because after that point the distribution doesn't change much, but the window keeps increasing. This could be a simple fix, but also might not need to be fixed due to the fact it only does it at high values, and high values are rarely used for the scale parameter.

One of the concerns with some of the user windows is the fact that the user doesn't know which sample vector belongs to which data set. This was in the first version of the User Data window but then was erased because it was thought that it wasn't needed. User input regarding the GUI will help us to determine the best way to proceed.

4.4 Future Work

There is still much work that can be done to both fix problems and enhance certain areas. The two main areas that still need work done to them is the parameter

estimation and the new extension, the 3D histogram plots. Much of the work in the future will probably focus on this new extension. The plans are now to smooth out the histograms and create best fit equations to each of the distributions. These equations will then eliminate the loading time, and will in turn make analysis faster. Other work that can be done would be to increase the capabilities of the tool. This would include adding more possible values for n and also including more distributions, possibly even those that contain shape parameters. With these new shape family distributions the parameter estimation tool would have to have its capabilities increased to estimate the shape parameter of a distribution. Also another procedure that was not used because of our new extension could be incorporated just as a way to double check the answers given by the new extension. This procedure is the one that is used in the algorithm currently and is based on the values of the α. Whichever distribution has the highest value for α then that is ruled as the true distribution. This may or may not be more accurate then our procedure, but only its implementation can give us the answer.

4.5 Acknowledgements

The author would like to extend his gratitude to Dr. Andrew Noga for his support and knowledge in the subject at hand. Dr. Noga also created the task for the summer, and continually renewed it when he saw fit. This allowed for continuous refinement and eventually the new extension which was the product of his vision of potential uses of the algorithm. Also, thanks are extended to Brett Smolenski for his suggestion of other PDF identification tests, and Mark Robertson for his insight into potential applications in video and imaging.

4.6 Tabular Values

Tabular values are the Monte Carlo runs of the $E(|Y_{i:n}|)$ and $|E(M_{i:n})|$, where the value is the *k*th largest deviate formal a standard Normal distribution, N(0,1), for a given sample size; or when preceded by a minus sign, the *k*th smallest deviate.

$E(\|Y_{i:n}\|)$						$\|E(M_{i:n})\|$					
k \ n	6	10	20	50	100	k \ n	6	10	20	50	100
1	1.3316	1.582	1.8922	2.2603	2.5137	1	1.2667	1.5389	1.8676	2.2484	2.5076
2	0.6764	1.0295	1.4263	1.8644	2.1535	2	0.6416	1.0015	1.4078	1.8546	2.1482
3	0.2885	0.6746	1.1458	1.637	1.9513	3	0.2017	0.656	1.1309	1.6286	1.9466
4		0.3951	0.9332	1.4712	1.8062	4		0.3756	0.9211	1.4636	1.8018
5		0.2096	0.7553	1.338	1.6915	5		0.1228	0.7454	1.331	1.6873
6			0.5982	1.2247	1.5953	6			0.5903	1.2182	1.5913
7			0.4545	1.1252	1.5118	7			0.4484	1.1193	1.5081
8			0.3226	1.0356	1.4377	8			0.3149	1.0302	1.4342
9			0.2102	0.9537	1.3708	9			0.187	0.9488	1.3674
10			0.1414	0.8777	1.3095	10			0.062	0.8731	1.3062
11				0.8065	1.2527	11				0.8023	1.2496
12				0.739	1.1997	12				0.7351	1.1968
13				0.6747	1.1498	13				0.6712	1.1469
14				0.613	1.1027	14				0.6099	1.1
15				0.5536	1.0578	15				0.5508	1.0552
16				0.4961	1.015	16				0.4935	1.0125
17				0.4401	0.9739	17				0.4378	0.9715
18				0.3854	0.9345	18				0.3834	0.9322
19				0.332	0.8964	19				0.3303	0.8942
20				0.2797	0.8596	20				0.278	0.8575
21				0.2289	0.8239	21				0.2264	0.8218

21	0.2289	0.8239	21	0.2264	0.8218
22	0.1807	0.7893	22	0.1755	0.7873
23	0.1379	0.7555	23	0.125	0.7537
24	0.105	0.7226	24	0.0749	0.7208
25	0.0868	0.6904	25	0.025	0.6887
26		0.659	26		0.6573
27		0.6281	27		0.6265
28		0.5979	28		0.5964
29		0.5682	29		0.5667
30		0.539	30		0.5376
31		0.5102	31		0.5089
32		0.4819	32		0.4807
33		0.454	33		0.4528
34		0.4264	34		0.4253
35		0.3991	35		0.398
36		0.372	36		0.3711
37		0.3453	37		0.3444
38		0.3188	38		0.318
39		0.2925	39		0.2919
40		0.2665	40		0.2658
41		0.2406	41		0.24
42		0.2149	42		0.2143
43		0.1896	43		0.1888
44		0.1646	44		0.1634
45		0.1404	45		0.138
46		0.1175	46		0.1129
47		0.097	47		0.0877
48		0.0797	48		0.0626
49		0.0673	49		0.0376
50		0.0608	50		0.0125

4.7 References

[1] Ozturk, A. An application of a distribution identification algorithm to signal detection problems. Signals, Systems and Computers, 1993. 1993 Conference Record of the Twenty-Seventh Asilomar Conference on 1-3 Nov. 1993 pp.248-252 vol. 1.

[2] Ozturk, A. and Dudewicz, E.J. A new statistical goodness-of-fit based on graphical representation. Biometrical Journal vol. 34 (4), pp.403-427, 1992.

[3] Ozturk, A. A general algorithm for univariate and multivariate goodness-of-fit tests based on graphical representation. Communications in statistics-theory and Methods. Vol.20 (10), pp.3111-3137, 1991.

[4] A. Ozturk. A new method for univariate and multivariate distribution identification. Pre-publication, pp.1-14, >1990.

[5] H. Leon Harter. Order Statistics and Their Uses in Testing and Estimation: Estimates based on Order Statistics of Samples from Various Populations. Aerospace Research Laboratories Vol 2, pp.425-426, 428, 436, 1970.

[6] Slamani, M.A.; Weiner, D.D.; Ozturk, A. A new approach to scene analysis for IR images. Circuits and Systems, 1994, Proceedings of the 37th Midwest Symposium on 3-5 Aug. 1994 pp.872-875 vol. 2.

[7] NIST/SEMATECH e-Handbook of Statistical Methods. Filliben, James J. 2005. NIST/SEMATECH. June 2005. <http://www.itl.nist.gov/div898/handbook/index.htm>.